

Research Article

A Fuzzy-Neural Ensemble and Geometric Rule Fusion Approach for Scheduling a Wafer Fabrication Factory

Hsin-Chieh Wu¹ and Toly Chen²

¹ Department of Industrial Engineering and Management, Chaoyang University of Science and Technology, Taiwan

² Department of Industrial Engineering and Systems Management, Feng Chia University, Taiwan

Correspondence should be addressed to Toly Chen; tcchen@fcu.edu.tw

Received 22 March 2013; Accepted 13 June 2013

Academic Editor: Pedro Ponce

Copyright © 2013 H.-C. Wu and T. Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this study, the fuzzy-neural ensemble and geometric rule fusion approach is presented to optimize the performance of job dispatching in a wafer fabrication factory with an intelligent rule. The proposed methodology is a modification of a previous study by fusing two dispatching rules and diversifying the job slacks in novel ways. To this end, the geometric mean of the neighboring distances of slacks is maximized. In addition, the fuzzy *c*-means (FCM) and backpropagation network (BPN) ensemble approach was also proposed to estimate the remaining cycle time of a job, which is an important input to the new rule. A new aggregation mechanism was also designed to enhance the robustness of the FCM-BPN ensemble approach. To validate the effectiveness of the proposed methodology, some experiments have been conducted. The experimental results did support the effectiveness of the proposed methodology.

1. Introduction

The scheduling of complex manufacturing systems is usually a NP-hard problem (see Table 1), which means it is very difficult for the production controller to find the best schedule within a reasonable period of time. To resolve this problem, a viable strategy is to develop a digital manufacturing model for the complex manufacturing system, based on which production simulation can be carried out to search for the optimal schedule in an efficient way. Based on this viewpoint, this study attempts to optimize the performance of job dispatching in a wafer fabrication factory, with the aid of the digital manufacturing model of the wafer fabrication factory. In this field, a digital manufacturing model is mainly used for three purposes—to evaluate the performance of a scheduling method, to optimize the scheduling performance, and to generate some test data for the subsequent optimization.

A wafer fabrication factory is a very complex manufacturing system featured by thousands of machines, tens of product types, various priorities, reentrant process flows, uncertain demand, and others. In addition, building a wafer fabrication factory usually needs billions of dollars. The capital-intensive feature makes the efficient use of a wafer fabrication

factory through good production management a very important task. However, some studies [1–3] noted that job scheduling in a semiconductor manufacturing factory is a very difficult task. As a result, many wafer fabrication factories suffer from lengthy cycle times. It is therefore not possible to promise their customers an attractable due date. On the other hand, owing to the uncertainty of the cycle time, the risk of missing the due date is also high.

Many different methods can be used for job dispatching in a wafer fabrication factory, such as dispatching rules, heuristics, data mining-based approaches [4, 5], agent technologies [4, 6–8], and simulation. Gupta and Sivakumar [9] classified the approaches for scheduling a wafer fabrication system into four categories: heuristic rules, mathematical, programming techniques, neighborhood search methods, and artificial intelligence techniques. A lot of studies have committed to the development of advanced dispatching rules for a wafer fabrication factory. For example, Altendorfer et al. [10] proposed the work in parallel queue (WIPQ) rule to maximize the throughput at a low level of work in process (WIP). Zhang et al. [11] proposed the dynamic bottleneck detection (DBD) approach, in which workstations are classified into

TABLE 1: The complexity of a decision problem.

Complexity	Meaning
P	A decision problem that is polynomial solvable.
NP	A decision problem that can be verified by a polynomial algorithm if a proper clue is given.
NP-hard	A decision problem P is called NP-hard if the entire class NP polynomially reduces to problem P.
NP-complete	A decision problem P is called NP-complete if it is both in classes NP and NP-hard.

several categories and then different dispatching rules, including first-in-first out-(FIFO), the shortest processing time until the next bottleneck (SPNB), and critical ratio (CR), are applied to these categories. Cao et al. [12] proposed a drum-buffer-rope based scheduling method for semiconductor manufacturing systems, which was focused on the control of the bottleneck machines. According to Lee et al. [13], the past experiences of scheduling were emphasized, and Petri nets were used to model the semiconductor manufacturing activities. As a result, the scheduling of a semiconductor manufacturing process was based on the token movements in the corresponding Petri net. Wang et al. [14] considered the scheduling of unrelated parallel machines in semiconductor manufacturing. After the problem reduction, some local heuristics could be proposed. Chen [15] smoothed the fluctuation in the estimated remaining cycle time, balanced it with that of the mean release rate, and proposed the nonlinear fluctuation smoothing policy for mean cycle time (NFSMCT). Hu et al. [16] divided the process flow into several stages and protected the bottleneck step at each reentrant stage from the system fluctuations. Although these dispatching rules are relatively easier to use, they cannot produce optimal or near-optimal scheduling results. To this end, a scheduling problem has to be formulated as a mathematical programming problem. The optimal solution of the mathematical programming problem gives the optimal scheduling of the manufacturing system. However, the mathematical programming problem of scheduling a wafer fabrication factory is too large and can hardly be solved. To resolve this problem, several treatments have been taken in the literature.

The first way is to choose the most suitable dispatching rule from the existing rules for the wafer fabrication factory dynamically. Chern and Liu [17] proved that general family-based scheduling rules are better than the individual job scheduling rule in terms of machine utilization for multi-server and multiple-job reentrant production systems under some conditions. Hsieh et al. [5] chose one rule from the fluctuation smoothing policy for mean cycle time (FSMCT), the fluctuation smoothing policy for cycle time variation (FSVCT), largest deviation first (LDF), one step ahead (OSA), and FIFO. The selection was based on the results of extensive production simulation.

The second way is to add adjustable parameters that can be optimized to the dispatching rule. For example, Chen [18] proposed the one-factor tailored NFSMCT (1f-TNFSMCT)

rule and the one-factor tailored nonlinear FSVCT (1f-TNFSVCT) rule. Various values of the parameter were tried in production simulation, and the one giving the best performance was chosen. Chen [19] used more parameters and proposed 2f-TNFSMCT and 2f-TNFSVCT. Both Dabbas and Fowler [20] and Dabbas et al. [21] combined some dispatching rules into a single rule by forming their linear combination with relative weights. However, a systematic procedure to determine the weights of these rules was missing. Similarly, Chiang et al. [22] developed a genetic algorithm (GA) for generating good dispatching rules through combining multiple rules linearly. Chen and Wang [23] proposed a biobjective nonlinear fluctuation smoothing rule with an adjustable factor (1f-biNFS) to optimize both the average cycle time and cycle time standard deviation at the same time. More degrees of freedom seem to be helpful for the performance of customizable rules. For this reason, Chen et al. [24] extended 1f-biNFS to a biobjective fluctuation smoothing rule with four adjustable factors (4f-biNFS). One drawback of these methods is that only static factors are used, and they must be determined in advance. To this end, most studies (e.g., [15–20]) performed extensive simulations. This is not only time consuming but also fails to consider enough possible combinations of these factors. To solve this problem, Kim et al. [25] suggested three techniques for accelerating rule comparison using production simulation.

The third way is to estimate the best schedule from a limited simulation results. In this manner, only a few combinations of the adjustable parameters will be tried in the production simulation, and then some estimation technique is applied to estimate the scheduling performance from the parameter values. To this end, Dabbas et al. [26] applied the design of experiment (DOE) techniques as well as the desirability function approach. In Li and Sigurdur [27], historic schedules were transformed into appropriate data files that were mined in order to find out which past scheduling decisions corresponded to the best practices. Harrath et al. [7] proposed a hybrid genetic algorithm (GA) and data mining approach to determine the optimal scheduling plan of a job shop, in which GA was used to generate a learning population of good solutions. These good solutions were then mined to find out some decision rules that could be transformed into a metaheuristic. Koonce and Tsai [4] proposed a similar methodology. Chen [28] attempted to relate the scheduling performance to the parameter values using a backpropagation network (BPN). However, the explanatory ability of the BPN was not as good as expected. Recently, Chen [29] constructed a nonlinear programming (NLP) model to optimize the parameter values in 2f-TNFSMCT and 2f-TNFSVCT by maximizing the standard deviation of slacks, which was considered to reduce possible ties. However, the NLP model was very difficult to optimize and required a lot of time to solve.

In short, the existing approaches have the following problems.

- (1) A more effective approach to optimize the parameter values is needed.

TABLE 2: The differences between the proposed methodology and the previous methods.

Method	Way of fusing rules	Estimating the remaining cycle time	Optimization method	Heuristic	Digital manufacturing model is used for	Required evaluation time
Hsieh et al. [5]	NA	NA	NA	Simulation	Evaluating/comparing scheduling performances	Lengthy
Dabbas et al. [26]	Linear	NA	DOE + the desirability function	Simulation	Evaluating/comparing scheduling performances	Lengthy
Harrath et al. [7]	Linear	NA	Enumeration	Association rule	Evaluating/comparing scheduling performances	Lengthy
Chen [29]	Nonlinear	FCM-BPN	Maximize the standard deviation of slacks	NLP	Confirmation study	Moderate
The proposed methodology	Nonlinear	FCM-BPN ensemble	Maximizing the geometric mean of the neighboring distances of slacks	NLP + systematic procedure	Confirmation study	Short

- (2) Maximizing the standard deviation of slacks may lead to the situation that most slacks concentrate on the two extremes.
- (3) How to avoid carrying out large-scale, time-consuming production simulation experiment is worth exploring.
- (4) New applications of the digital manufacturing model are still to be discovered.

In order to solve some of these problems and to further improve the performance of job scheduling in a wafer fabrication factory, Chen's approach has been modified, and a fuzzy-neural-ensemble and geometric rule fusion approach was proposed in this study. The proposed methodology has the following new characteristics.

- (1) To diversify the slack, the geometric mean of the neighboring distances of slacks is to be considered, rather than the standard deviation.
- (2) Diversifying the slacks leads to a NLP problem, which is not easy to solve. To solve this problem, a systematic procedure has been established to search the optimal solution of the NLP problem in an effective manner.
- (3) The fuzzy c-means (FCM) and backpropagation network (BPN) ensemble approach [30] was also modified to estimate the remaining cycle time of a job. With a novel aggregation mechanism, the FCM-BPN ensemble approach is not only accurate but also robust to the untrained data.
- (4) The two existing rules, 2f-TFSMCT and 2f-TNFSVCT, are fused in a novel way so that two objectives, the average cycle time and cycle time variation, can be improved at the same time. Data fusion is a technique that gathers information and combines data from multiple sources in order to achieve inferences. This method is more efficient and potentially more accurate than if the inference was achieved from a single source.

The differences between the proposed methodology and the previous methods are summarized in Table 2.

The outline of this paper is described as follows. First, in Section 2, a new rule is proposed through the fusion of the two rules, 2f-TNFSMCT and 2f-TNFSVCT, in a novel way. To determine the values of parameters in the new rule, a NLP model is established. A systematic procedure is then used to solve the NLP problem. In addition, the remaining cycle time of each job is a necessary input of the new rule. The FCM and BPN ensemble approach, therefore, is applied to estimate the remaining cycle time of a job. In Section 3, the performance of the proposed methodology is evaluated by carrying out some experiments. Finally, this paper is concluded in Section 4.

2. Methodology

The variables and parameters that will be used in this study are defined in the Abbreviations section.

The proposed methodology includes the following six steps.

Step 1. Fuse the two rules, 2f-FSMCT and 2f-FSVCT, to form the new rule 2f-biFS.

Step 2. Establish a NLP model to optimize the parameters of the new rule.

Step 3. Apply a systematic procedure to solve the NLP problem.

Step 4. Estimate the remaining cycle time of a job using the FCM-BPN ensemble approach.

Step 5. Incorporate the estimated remaining cycle time into the new rule.

Step 6. Carry out experiments to evaluate the performance of the new rule.

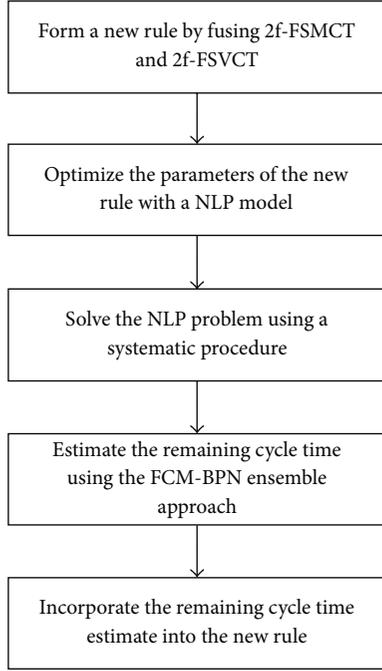


FIGURE 1: The flow chart of the proposed methodology.

The flow chart of the proposed methodology is shown in Figure 1. The algorithm proposed in this study is therefore indicated with FCM-BPN-ensemble-2f-biFS. Some steps correspond to the operations of the digital manufacturing model, as summarized in Table 3.

2.1. The Wafer Fabrication Environment. In this study, a wafer fabrication factory located in Taichung Scientific Park of Taiwan is analyzed. The wafer fabrication factory has a monthly capacity of about 25,000 wafers. Currently, more than 10 types of memory products are being produced with 58 nm~110 nm technologies in the wafer fabrication factory. For this, more than 500 workstations are used. Machine failure and repairing times basically follow the exponential and uniform distributions. Jobs released into the fabrication factory are assigned three types of priorities, that is, “normal”, “hot” and “super hot”. Jobs with the highest priorities will be processed first. The current scheduling policy in the wafer fabrication factory is FIFO. As a result, the longest average cycle time exceeds three months with a variation of more than 300 hours. The wafer fabrication factory is therefore seeking better scheduling rules to replace FIFO.

2.2. The New Rule. Information fusion has been widely used in scheduling and other system control purposes. According to Zhao et al. [31], the historical data of passenger flow of urban rail transportation were gathered and fused, in order to calculate the characteristic data for real-time schedule. Leuret et al. [32] presented an intelligence fusion system based on gain scheduling control and hybrid neural networks for system fault detection and fault-tolerant control. Li et al. [33] measured and fused the collected raw data for the scheduling

TABLE 3: Operations of the digital manufacturing model in the proposed methodology.

Step	Operation of the digital manufacturing model
The new rule	Control
FCM	Reduction
BPN	Approximation
The aggregation mechanism	Combining submodels

of a multihop hierarchical sensor network. In this study, the two basic fluctuation smoothing (FS) rules [34] are to be fused, in order to have a better control over the scheduling of the wafer fabrication factory. The first one is FSMCT, in which the slack of a job is defined as

$$SKM_{ju} = \frac{j}{\lambda} - RCTE_{ju} \quad (1)$$

which is aimed at the reduction of the average cycle time. Reducing the cycle time is a critical task in managing a wafer fabrication factory, especially, in a mass production setting such as in a memory fabrication factory [35]. The second rule is FSVCT:

$$SKV_{ju} = R_j - RCTE_{ju} \quad (2)$$

which is aimed at the minimization of cycle time standard deviation. Some variants of the two rules were summarized in Table 4 [18, 19, 23, 36, 37]. Jobs with smaller slack values (SKM_{ju} or SKV_{ju}) will be processed earlier. However, a tie is formed if more than one job has the same slack, which results in scheduling difficulties.

Chen [19] normalized the parameters in the FS rules and then divided them, which led to the 2f-TNFSMCT rule:

$$SKM_{ju} = \left(\frac{\lambda\beta}{(n-1)(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \cdot \left(\frac{j}{\lambda} - RCTE_{ju} + \zeta \left(RCTE_{ju} - \frac{1}{\lambda} \right) \right) \quad (3)$$

and the 2f-TNFSVCT rule:

$$SKV_{ju} = \left(\frac{\beta}{\alpha(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \cdot (R_j - RCTE_{ju} + \zeta(RCTE_{ju} - \min(R_j))), \quad (4)$$

where

$$\alpha = \max(R_j) - \min(R_j) \quad (5)$$

$$\beta = \max(RCTE_{ju}) - \min(RCTE_{ju})$$

$0 \leq \xi, \zeta \leq 1$. In Chen’s study, three models were established to form the combination of ξ and ζ . For example,

$$\text{(linear model)} \quad \xi = \zeta, \quad (6)$$

$$\text{(nonlinear model)} \quad \xi = \zeta^k, \quad k \geq 0, \quad (7)$$

$$\text{(logarithmic model)} \quad \xi = \frac{\ln(1 + \zeta)}{\ln 2}. \quad (8)$$

TABLE 4: Some variants of FSMCT and FSVCT.

Rule	
NFS [23]	$SK_{ju} = \frac{((R_j - \min(R_j)) / (\max(R_j) - \min(R_j)))^\xi \cdot ((j-1) / (n-1))^{1-\xi}}{(RCTE_{ju} - \min(RCTE_{ju})) / (\max(RCTE_{ju}) - \min(RCTE_{ju}))}$
1f-TNFSMCT [18]	$SKV_{ju} = \left(\frac{\lambda\beta}{(n-1)(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \cdot \left(\frac{j}{\lambda} - RCTE_{ju} + \xi \left(RCTE_{ju} - \frac{1}{\lambda} \right) \right)$
1f-TNFSVCT [18]	$SKM_{ju} = \left(\frac{\beta}{\alpha(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \cdot (R_j - RCTE_{ju} + \xi(RCTE_{ju} - \min(R_j)))$
2f-TNFSMCT [19]	$SKV_{ju} = \left(\frac{\lambda\beta}{(n-1)(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \cdot \left(\frac{j}{\lambda} - RCTE_{ju} + \zeta \left(RCTE_{ju} - \frac{1}{\lambda} \right) \right)$
2f-TNFSVCT [19]	$SKM_{ju} = \left(\frac{\beta}{\alpha(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \cdot (R_j - RCTE_{ju} + \zeta(RCTE_{ju} - \min(R_j)))$

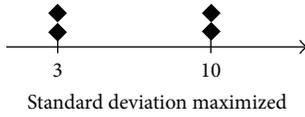


FIGURE 2: The results of maximizing the standard deviation of slacks.

However, no model can guarantee the optimization of the scheduling performance. To solve this problem, Wang et al. [37] advocated that, through diversifying the slacks of jobs, the overlapping of slacks can be reduced, which avoids mis-scheduling and is conducive to the scheduling performance. To this end, Wang maximized the standard deviation of slacks. However, this treatment may lead to the situation that most slacks concentrate on the two extremes [29], as illustrated in Figure 2.

In this study, to diversify the slack, the geometric mean of the neighboring distances of slacks is maximized instead:

$$\text{Max } Z = \sqrt[n-1]{\prod_{j=2}^n (SK_{(j)u} - SK_{(j-1)u})}. \quad (9)$$

An example is illustrated in Figure 3. Obviously, to maximize the geometric mean of the neighboring distances, the slacks need to be evenly distributed.

However, (9) is not easy to solve. To tackle this, the following procedure is established.

- (1) Change the values of parameters.
- (2) Derive the slack of each job.
- (3) Exclude jobs with very large or small slacks.
- (4) Sort the slacks of the remaining jobs.
- (5) Calculate the distance between each neighboring pair.
- (6) Add a small value Δ , for example, 0.01, to the distance to avoid the case of zero slack.
- (7) Obtain the geometric mean of the distances.
- (8) If the geometric mean has been maximized, stop; otherwise, return to Step 1.



FIGURE 3: The results of maximizing the geometric mean of the neighboring distances of slacks.

As a result, in 2f-TNFSMCT, the following objective function is to be optimized:

$$\text{Max } Z_1 = \sqrt[n-1]{\prod_{j=2}^n (SKM_{(j)u} - SKM_{(j-1)u} + \Delta)}, \quad (10)$$

where $SKM_{(j)u}$ is the slack of the job that ranks the j th according to 2f-TNFSMCT. On the other hand, in 2f-TNFSVCT, the following objective function is to be optimized:

$$\text{Max } Z_2 = \sqrt[n-1]{\prod_{j=2}^n (SKV_{(j)u} - SKV_{(j-1)u} + \Delta)}, \quad (11)$$

where $SKV_{(j)u}$ is the slack of the job that ranks the j th according to 2f-TNFSVCT. To fuse the two rules into a single one, that is, 2f-biFS, a natural way is to maximize the weighted geometric mean of the two objective functions, subject to several linear and nonlinear constraints:

$$\text{Max } Z_3 = Z_1^\omega Z_2^{1-\omega} \quad (12)$$

subject to

$$Z_1 = \sqrt[n-1]{\prod_{j=2}^n (SKM_{(j)u} - SKM_{(j-1)u} + \Delta)}$$

$$Z_2 = \sqrt[n-1]{\prod_{j=2}^n (SKV_{(j)u} - SKV_{(j-1)u} + \Delta)}$$

$$\begin{aligned}
SKM_{ju} &= \left(\frac{\lambda\beta}{(n-1)(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \\
&\quad \cdot \left(\frac{j}{\lambda} - RCTE_{ju} + \zeta \left(RCTE_{ju} - \frac{1}{\lambda} \right) \right) \\
SKV_{ju} &= \left(\frac{\beta}{\alpha(RCTE_{ju} - \min(RCTE_{ju}))} \right)^\xi \\
&\quad \cdot (R_j - RCTE_{ju} + \zeta(RCTE_{ju} - \min(R_j))) \\
&\quad \xi = \zeta,
\end{aligned} \tag{13}$$

$$\xi = \zeta^k, \quad k \geq 0, \tag{14}$$

or

$$\begin{aligned}
\xi &= \frac{\ln(1 + \zeta)}{\ln 2} \\
0 &\leq \xi, \quad \zeta \leq 1.
\end{aligned} \tag{15}$$

This model is obviously a NLP problem. Since ξ and ζ are limited to be within $[0, 1]$, the following exhaustive search algorithm can be used to reach the optimal solution of the NLP problem in an effective way.

Step 1. Choose a model from (6)~(8).

Step 2. Determine the weight ω .

Step 3. Let $\xi = 0$, $Z_{\max} = 0$

Step 4. Calculate ζ .

Step 5. Calculate the slack of each job according to 2f-TNFSMCT.

Step 6. Calculate Z_1 .

Step 7. Calculate the slack of each job according to 2f-TNFSVCT.

Step 8. Calculate Z_2 .

Step 9. Calculate Z_3 . If Z_3 is greater than Z_{\max} , then $Z_{\max} = Z_3$.

Step 10. $\xi = \xi + 0.01$. If ξ is greater than 1, go to Step 11; otherwise, return to Step 4.

Step 11. Search the neighborhood of the current optimal solution to fine-tune the value of ξ .

An example is provided in Table 5 to illustrate the procedure mentioned above, in which $\alpha = 734$, $\beta = 1836$, and $\lambda = 1.18$. If the nonlinear model in (7) is referred to and $k = 2$; $\omega = 0.6$, then the optimal objective function value is $Z_3^* = 47.1$ when $(\xi^*, \zeta^*) = (0.000016, 0.004)$.

TABLE 5: A demonstrative example.

#	R_j (hrs)	j	$RCTE_{ju}$ (hrs)
1	102	159	1399
2	756	37	1127
3	826	37	1223
4	652	86	1822
5	208	55	530
6	783	84	2040
7	800	96	2366
8	478	52	942
9	469	65	1116
10	699	32	995
11	836	85	2151
12	497	45	883
13	596	101	2047
14	798	34	1146
15	197	79	743
16	804	85	2092
17	163	78	647
18	457	44	810
19	523	100	1851

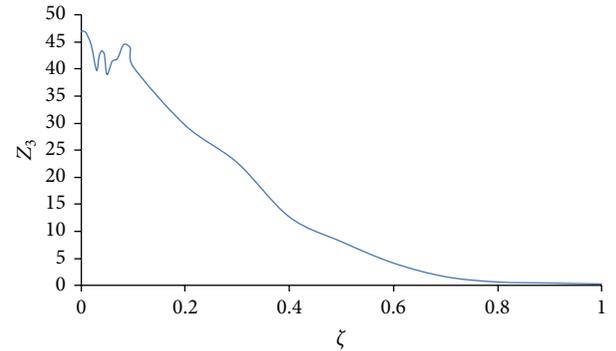


FIGURE 4: The objective function values associated with various values of ζ .

A parametric analysis was also performed. The objective function values associated with various values of ζ ($\xi = \zeta^2$) are shown in Figure 4. In theory, Z_3 converges to a small positive value (Δ).

The remaining cycle time of a job is an important input to the 2f-biFS rule. To estimate it, the FCM-BPN ensemble approach [30] was modified, and a new aggregation mechanism was proposed, as described in the next section.

2.3. The FCM Approach. FCM has been applied to a variety of purposes in a wafer fabrication factory. For instance, Liu and Chen [38] proposed a modified FCM algorithm along with a quality index (ψ) to cluster the characteristic values of low-yield wafers. Wang [39] proposed a hybrid scheme combining the entropy FCM (EFCM) with spectral clustering to denoise the noisy wafer map and to extract meaningful defect clusters. In the proposed methodology, the remaining cycle

times of all jobs in front of a machine must be estimated before determining the order of these jobs. In addition, job classification is considered to be beneficial to the accuracy of the remaining cycle time estimation [30, 36–39]. For these reasons, in the FCM-BPN ensemble approach, jobs are classified into K clusters using FCM. The use of FCM has the following advantages.

- (1) Classification is a subjective concept. Therefore, the absolute job classification may not be correct. In a fuzzy classification method like FCM, a job belongs to more than one cluster with different degrees, which provides a solution to this problem. The FCM-BPN ensemble approach considers the estimates from all of the clusters and may be more robust than the FCM-BPN without ensemble.
- (2) In a crisp clustering method, it is possible that some clusters have a very few examples. In contrast, in FCM all examples belong to a cluster with different degrees, which provides a solution to this problem.

The objective function of FCM is to minimize the weighted sum of squared error:

$$\text{Min} \sum_{k=1}^K \sum_{j=1}^n \mu_{j(k)}^m e_{j(k)}^2, \quad (16)$$

where K is the required number of clusters; n is the number of jobs; $\mu_{j(k)}$ indicates the membership that job j belongs to cluster k ; $e_{j(k)}$ measures the distance from job j to the centroid of cluster k ; $m \in [1, \infty)$ is a parameter to adjust the fuzziness and is usually set to 2. The procedure of FCM starts from the normalization of data. The (normalized) attributes of job j are placed in vector $\mathbf{x}_j = [x_{jp}]$. Subsequently, an initial guess of the clustering results is generated. The performance of FCM is highly sensitive to the initial guess. After each round of clustering, the centroid of each cluster is updated as follows:

$$\begin{aligned} \bar{x}_{(k)} &= \{\bar{x}_{(k)p}\}; \quad p = 1 \sim P, \\ \bar{x}_{(k)p} &= \frac{\sum_{j=1}^n \mu_{j(k)}^m x_{jp}}{\sum_{j=1}^n \mu_{j(k)}^m} \end{aligned} \quad (17)$$

in which

$$\mu_{j(k)} = \frac{1}{\sum_{q=1}^K (e_{j(k)}/e_{j(q)})^{2/(m-1)}}, \quad (18)$$

$$e_{j(k)} = \sqrt{\sum_{p=1}^P (x_{jp} - \bar{x}_{(k)p})^2}, \quad (19)$$

where $\bar{x}_{(k)}$ is the centroid of cluster k . $\mu_{j(k)}$ and $e_{j(k)}$ are also updated at the same time. The iterative process of job clustering continues until the clustering results converge:

$$\max_k \max_j |\mu_{j(k)}^{(t)} - \mu_{j(k)}^{(t-1)}| < d, \quad (20)$$

where d is a real number representing the threshold for the convergence of membership. A problem of FCM is that

the number of clusters must be decided in advance. To this end, the separate distance test (S test) proposed by Xie and Beni [40] is applicable:

$$S = \frac{J_m}{n \times e_{\min}^2}, \quad (21)$$

where

$$J_m = \sum_{k=1}^K \sum_{j=1}^n \mu_{j(k)}^m e_{j(k)}^2, \quad (22)$$

$$e_{\min}^2 = \min_{k_1 \neq k_2} \left(\sum_{\text{all } p} (\bar{x}_{(k_1)p} - \bar{x}_{(k_2)p})^2 \right), \quad (23)$$

$$K \in \mathbb{Z}^+. \quad (24)$$

K is the number of categories. According to (22) and (21), J_m is a function of K , and S is a function of J_m . So, we can try several values of K to minimize S . In fact, the K value minimizing S determines the suitable number of clusters. After FCM, the original digital manufacturing model has been split into several smaller ones that are easier to deal with.

An example is provided in Table 6 to illustrate the application of FCM. All decision and response variables have been normalized into [0.1 0.9] to consider the situation that the future value may be greater/smaller than all the historical values. The Fuzzy Logic Toolbox of MATLAB is used to implement the FCM approach.

The results of the S test were summarized in Table 7. In this case, the optimal number of clusters was 3. The clustering results when $K = 3$ are shown in Table 8. If each job is classified into the cluster to which the membership is the highest, then the classification results are as shown in Table 9.

2.4. The BPN Approach. After clustering, a BPN is constructed for each cluster. A portion of the jobs in each cluster is input as the “training examples” to the BPN to determine the parameter values. The configuration of the BPN is established as follows. The BPN is a three-layer multiple-input single-output (MISO) system. Inputs are the normalized values of the P parameters associated with a job. There is only one hidden layer with $2P$ neurons. The output from the BPN is the normalized value of the remaining cycle time estimate. The activation function used in each layer is Log Sigmoid function (see Figure 5). The Levenberg-Marquardt algorithm is applied to train the BPN because of its efficiency. BPN training using the Levenberg-Marquardt algorithm has been stated in many past studies in this field [29] and therefore will not be repeated here.

In the previous example, take cluster 1 as an example. There are 16 jobs in this cluster. Split them into two parts: the training data (the first 12 jobs) and the testing data (the remaining jobs) and then construct a BPN to estimate the cycle time of jobs in this cluster. The Neural Network Toolbox of MATLAB is used to implement the BPN approach. The estimation results were summarized in Table 10. The estimation performance was basically very good. For the testing data, however, the estimation error was relatively large.

TABLE 6: An illustrative example.

j	u	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	RCTE $_{ju}$
1	197	0.300	0.485	0.900	0.367	0.367	0.162	0.667
2	192	0.500	0.604	0.900	0.367	0.367	0.362	0.698
3	188	0.300	0.248	0.100	0.456	0.500	0.163	0.758
4	202	0.500	0.159	0.900	0.100	0.233	0.203	0.738
5	187	0.500	0.604	0.100	0.544	0.567	0.349	0.675
6	184	0.500	0.396	0.900	0.722	0.700	0.279	0.703
7	197	0.300	0.426	0.900	0.633	0.367	0.314	0.719
8	184	0.300	0.574	0.900	0.811	0.567	0.314	0.727
9	178	0.500	0.811	0.900	0.900	0.633	0.314	0.670
10	212	0.300	0.752	0.900	0.633	0.567	0.106	0.783
11	177	0.300	0.900	0.900	0.722	0.633	0.106	0.697
12	186	0.300	0.426	0.900	0.544	0.500	0.106	0.736
13	199	0.300	0.396	0.900	0.456	0.567	0.106	0.713
14	195	0.300	0.159	0.900	0.544	0.567	0.100	0.806
15	223	0.500	0.189	0.100	0.278	0.367	0.291	0.858
16	223	0.300	0.781	0.100	0.456	0.300	0.168	0.900
17	206	0.100	0.722	0.900	0.189	0.100	0.105	0.722
18	212	0.100	0.367	0.900	0.367	0.433	0.105	0.833
19	168	0.500	0.544	0.900	0.278	0.167	0.351	0.614
20	200	0.500	0.426	0.900	0.278	0.167	0.351	0.678
21	140	0.100	0.841	0.900	0.367	0.167	0.142	0.445
22	156	0.300	0.130	0.100	0.456	0.167	0.162	0.548
23	206	0.300	0.396	0.900	0.633	0.633	0.123	0.820
24	152	0.300	0.515	0.900	0.811	0.767	0.123	0.506
25	187	0.100	0.544	0.100	0.722	0.700	0.123	0.633
26	185	0.500	0.781	0.900	0.722	0.433	0.319	0.634
27	137	0.500	0.663	0.900	0.722	0.567	0.260	0.423
28	151	0.700	0.396	0.900	0.811	0.633	0.620	0.486
29	148	0.500	0.100	0.900	0.722	0.567	0.250	0.402
30	173	0.300	0.426	0.900	0.722	0.700	0.216	0.667
31	161	0.300	0.307	0.900	0.722	0.767	0.158	0.486
32	108	0.500	0.396	0.900	0.811	0.700	0.264	0.155
33	115	0.300	0.870	0.900	0.900	0.900	0.302	0.164
34	100	0.300	0.485	0.900	0.633	0.633	0.337	0.150
35	97	0.900	0.396	0.900	0.900	0.633	0.900	0.100

The estimation accuracy can be evaluated with the following indexes:

$$\begin{aligned}
 \text{MAE} &= \frac{\sum_{j=1}^n |CT_j - \text{CTE}_j|}{n}, \\
 \text{MAPE} &= \frac{\sum_{j=1}^n |CT_j - \text{CTE}_j| / CT_j}{n} \cdot 100\%, \quad (25) \\
 \text{RMSE} &= \sqrt{\frac{\sum_{j=1}^n (CT_j - \text{CTE}_j)^2}{n}}.
 \end{aligned}$$

a_j and f_j denote the actual value and forecast of job j , respectively; n is the total number of data. In this example,

- MAE = 43 (hrs),
- MAPE = 3.7%,
- RMSE = 62 (hrs).

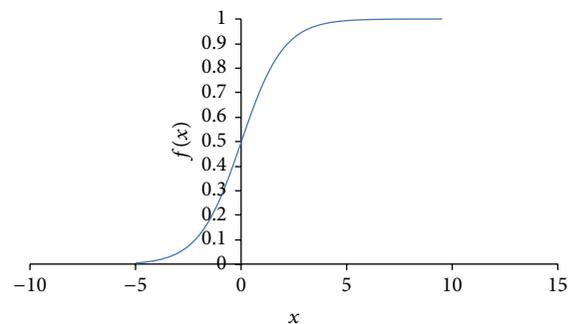


FIGURE 5: The Log Sigmoid function.

The performance of the FCM-BPN approach was also compared with those of BPN trained using the Levenberg-Marquardt algorithm and FCM-BPN trained using the gradient descent algorithm. The results are shown in Table II.

TABLE 7: The results of the S test.

Number of clusters (K)	J_m	e_{\min}^2	S
2	0.0739	0.0027	0.7946
3	0.0444	0.0028	0.4586
4	0.0316	0.0014	0.6298
5	0.0247	0.0014	0.4882
6	0.0205	0.0009	0.6336
7	0.0175	0.0005	1.1102
8	0.0154	0.0003	1.2577

TABLE 8: The clustering results by FCM.

j	$\mu_{j(1)}$	$\mu_{j(2)}$	$\mu_{j(3)}$
1	0.91	0.061	0.03
2	0.698	0.21	0.092
3	0.06	0.046	0.894
4	0.559	0.22	0.221
5	0.118	0.118	0.764
6	0.262	0.664	0.073
7	0.721	0.218	0.062
8	0.298	0.629	0.073
9	0.244	0.654	0.101
10	0.497	0.396	0.107
11	0.365	0.508	0.127
12	0.802	0.152	0.047
13	0.766	0.173	0.061
14	0.567	0.284	0.149
15	0.139	0.097	0.764
16	0.186	0.137	0.678
17	0.574	0.224	0.202
18	0.691	0.179	0.13
19	0.635	0.22	0.145
20	0.649	0.202	0.149
21	0.521	0.296	0.182
22	0.164	0.122	0.714
23	0.546	0.353	0.101
24	0.173	0.766	0.061
25	0.18	0.195	0.624
26	0.355	0.55	0.095
27	0.134	0.828	0.038
28	0.264	0.6	0.137
29	0.332	0.527	0.141
30	0.275	0.661	0.064
31	0.253	0.659	0.088
32	0.212	0.682	0.106
33	0.246	0.608	0.146
34	0.26	0.632	0.107
35	0.295	0.487	0.218

2.5. *Aggregating the Estimates from BPNs.* In past studies, a job is usually classified into the cluster with the highest membership. However, that makes FCM not different from crisp classifiers. To tackle this problem, Chen [30] applied the BPNs of all clusters to estimate the cycle time of a job and used a BPN to aggregate these estimates. However, the BPN aggregator is very sensitive and may lead to unexpected results for untrained data. In addition, Chen [30] also showed that the aggregation performance of simple linear combination was

TABLE 9: The classification results (the highest membership principle).

Cluster	Jobs
1	6, 8, 9, 10, 11, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34
2	3, 5, 15, 16, 22, 25
3	1, 2, 4, 7, 12, 13, 17, 18, 19, 20, 21

TABLE 10: The estimation results (cluster 1).

j	CT_j	CTE_j	$ CT_j - CTE_j $
6	1227	1226	1.6
8	1215	1230	15.3
9	1228	1231	2.5
10	1266	1260	5.4
11	1285	1268	17.0
23	1280	1238	42.1
24	1286	1258	28.0
26	1214	1224	9.6
27	1251	1227	23.9
28	1222	1193	29.4
29	1187	1204	17.5
30	1205	1235	30.4
31	1120	1240	120.1
32	1133	1220	88.0
33	1130	1263	133.2
34	1113	1238	124.5

TABLE 11: Comparison of the estimation performances of various approaches.

Performance measure	BPN (the Levenberg-Marquardt algorithm)	FCM-BPN (the gradient descent algorithm)	FCM-BPN (the Levenberg-Marquardt algorithm)
MAE	47 (hrs)	79	43
MAPE	4.1%	6.7%	3.7%
RMSE	70 (hrs)	98	62

even worse. For these reasons, a new aggregation mechanism is proposed as follows.

According to (18), μ_{jk} is inversely proportional to $e_{j(k)}^{2/(m-1)}$:

$$\mu_{j(k)} \propto \frac{1}{e_{j(k)}^{2/(m-1)}}. \tag{26}$$

Therefore,

$$e_{j(k)} \propto {}^{2/(m-1)}\sqrt{\frac{1}{\mu_{j(k)}}}. \tag{27}$$

Further, according to (19), the error is proportional to the distance to the centroid. For this reason, a natural way to aggregate the estimates from the BPNs is

$$RCTE_{ju} = \frac{\sum_{k=1}^K {}^{2/(m-1)}\sqrt{1/\mu_j(k)} \cdot RCTE_{ju}(k)}{\sum_{k=1}^K {}^{2/(m-1)}\sqrt{1/\mu_j(k)}}, \tag{28}$$

where $RCTE_{ju}(k)$ is the remaining cycle time of job j estimated by the BPN of cluster k .

TABLE 12: The estimation results after aggregation (cluster 1).

j	CT_j	CTE_j	$ CT_j - CTE_j $
6	1227	1201	26.3
8	1215	1199	16.4
9	1228	1170	58.2
10	1266	1228	38.0
11	1285	1222	62.8
23	1280	1203	77.1
24	1286	1240	45.7
26	1214	1187	26.7
27	1251	1195	56.4
28	1222	1158	64.0
29	1187	1191	3.6
30	1205	1176	28.8
31	1120	1147	27.0
32	1133	1103	30.0
33	1130	1172	42.0
34	1113	1158	45.0

In the previous example, after aggregation, the estimation results are shown in Table 12. The estimation accuracy, measured in terms of MAE, for example, was slightly improved from 43 hours to 40.5 hours. It is worth noting that the estimation error was significantly reduced for the untrained (testing) data, which supported the robustness of the aggregation mechanism. In Table 13, the performances of some aggregation mechanisms were also compared. In linear combination, the weighted sum of the estimates from all of the BPNs was obtained, in which the weight was equal to the membership of belonging to a cluster. According to the experimental results, although the BPN aggregation mechanism achieved the best estimation accuracy, it may generate a considerable deviation for untrained (testing) data. In contrast, the proposed aggregation mechanism did not have this problem, and the estimation accuracy was also quite good.

3. Experiments

Twelve scheduling policies, including FIFO, EDD, CR, SRPT, FSVCT, FSMCT, DBD [11], NFS [23], 2f-TNFSMCT, 2f-TNFSVCT, the slack-diversifying rule [29], and the proposed FCM-BPN-ensemble-2f-biFS, were applied to schedule the target wafer fabrication factory. In total, the data of 1000 jobs of five major cases have been collected and were separated by their product types and priorities.

In FIFO, jobs were sequenced on each machine first by their priorities, then by their arrival times at the machine. In EDD, jobs were sequenced first by their priorities, then by their due dates that were equal to the multiple of the total processing times plus the release time:

$$DD_j = R_j + c \cdot TPT_j, \quad (29)$$

where c is called the cycle time multiplier and is determined based on the cycle time statistics. In CR, jobs were sequenced

TABLE 13: The performances of various aggregation mechanisms (cluster 1).

MAE (hrs)	Without aggregation	Linear combination	BPN	The proposed mechanism
Training data	18.6	60.0	12.7	42.0
Testing data	116.5	147.8	90.9	36.0
Both	43.0	81.9	32.3	40.5

first by their priorities, then by their critical ratios that are defined as

$$CR_j = \frac{DD_j - t}{RPT_{ju}}. \quad (30)$$

In SRPT, jobs were sequenced first by their priorities, then by their remaining processing times. In FSVCT and FSMCT, jobs were sequenced on each machine first by their priorities and then by their slack values, which were determined by (1) and (2). To assist this, the remaining cycle time statistics have been collected from the historical data. The operation of NFS is similar, in which the slack of a job is calculated in the following way:

$$SK_{ju} = \left(\frac{R_j - \min R_j}{\max R_j - \min R_j} \right)^\xi \cdot \left(\frac{j-1}{n-1} \right)^{1-\xi} \times \left(\frac{RCTE_{ju} - \min RCTE_{ju}}{\max RCTE_{ju} - \min RCTE_{ju}} \right)^{-1}, \quad (31)$$

where ξ ranges between 0 and 1 and will be optimized by trying various values in the experiment. In DBD, workstations are first classified into four categories, and then CR + FIFO, the shortest processing time until the next bottleneck (SPNB) + CR + FIFO, shortest processing time (SPT) + CR + FIFO, and CR + FIFO are applied to sequence jobs in these categories, respectively. In the slack-diversifying rule [29], the weighted sum of the standard deviations of slacks calculated by 2f-TNFSMCT and 2f-TNFSVCT was obtained, and then jobs were sequenced according to this value.

Two performance measures, the average cycle time and cycle time standard deviation of each case, achieved by the twelve scheduling policies were observed and summarized in Tables 14 and 15.

According to the experimental results, the following points can be made.

- (1) For the average cycle time, the proposed FCM-BPN-ensemble-2f-biFS outperformed the eleven existing policies. The most significant advantage was over the 2f-TNFSVCT rule, which is about 28% on average.
- (2) On the other hand, the proposed FCM-BPN-ensemble-2f-biFS also surpassed the existing rules, especially FSMCT, in reducing cycle time standard deviation. The average advantage was up to 61%.
- (3) Both the slack-diversifying rule [29] and the proposed FCM-BPN-ensemble-2f-biFS achieved very good

TABLE 14: The performances of various approaches in the average cycle time.

Avg. cycle time (hrs)	Case I	Case II	Case III	Case IV	Case V
FIFO	1254	400	317	1278	426
EDD	1098	347	302	1431	436
CR	1145	354	299	1501	443
SRPT	948	350	308	1737	457
FSMCT	1313	347	293	1851	470
FSVCT	1014	382	315	1672	475
NFS	1456	407	321	1452	421
DBD	1031	349	297	1501	463
2f-TNFSMCT	1369	379	306	1361	399
2f-TNFSVCT	1465	416	318	1551	500
The slack-diversifying rule	1076	289	269	1132	388
FCM-BPN-ensemble-2f-biFS	987	276	254	1121	376

TABLE 15: The performances of various approaches in cycle time standard deviation.

Cycle time standard deviation (hrs)	Case I	Case II	Case III	Case IV	Case V
FIFO	55	24	25	87	51
EDD	128	25	23	50	61
CR	70	30	18	57	52
SRPT	248	31	22	106	53
FSMCT	419	33	16	129	104
FSVCT	280	37	27	201	77
NFS	87	49	19	44	47
DBD	136	25	19	77	29
2f-TNFSMCT	75	37	17	47	19
2f-TNFSVCT	38	38	29	33	24
The slack-diversifying rule	86	26	15	54	21
FCM-BPN-ensemble-2f-biFS	83	21	16	55	19

scheduling performances, which supported slack diversifying to be a viable strategy for improving the performances of similar rules.

- (4) In addition, the performance of the proposed FCM-BPN-ensemble-2f-biFS was also better than the slack-diversifying rule [29]. This showed that the treatments taken in this study, including the new way of slack diversification and forming the BPN ensemble, were indeed effective. To differentiate their effects, additional experiments have been conducted, and the results were shown in Figure 6. Obviously, forming the BPN ensemble was more effective than the new way of slack diversification. That was reasonable because in theory slack diversification has its limits.
- (5) If the cycle time is long, the remaining cycle time will be much longer than the remaining processing time, which leads to the ineffectiveness of SRPT. As a result, SRPT performed poor in reducing the average cycle times of such cases, such as case IV. FSMCT has similar problems.
- (6) In contrast, the performances of EDD and CR were satisfactory for cases with short cycle times because it is less likely that the cycle time (multiplier) will deviate from the estimated value.

- (7) We also compared the performances of different fusion mechanisms. To this end, the linear fusion in which the slack values derived by the two rules were simply added was also tested. The results are shown in Figure 7. Obviously, the effects of the linear fusion mechanism on the average cycle time were poor, and certainly the proposed geometric fusion mechanism could be used as an alternative.

4. Conclusions and Directions for Future Research

To further improve the performance of job scheduling in a wafer fabrication factory, the slack-diversifying rule used by Chen [29] is modified, and the FCM-BPN-ensemble-2f-biFS was proposed in this study. In the proposed methodology, two existing rules, 2f-TNFSMCT and 2f-TNFSVCT, are fused by maximizing the geometric mean of the neighboring distances of slacks. In addition, to enhance the accuracy of estimating the remaining cycle time, the FCM-BPN ensemble approach is applied with a novel aggregation mechanism that considers the operations of FCM and will not overreact for untrained data.

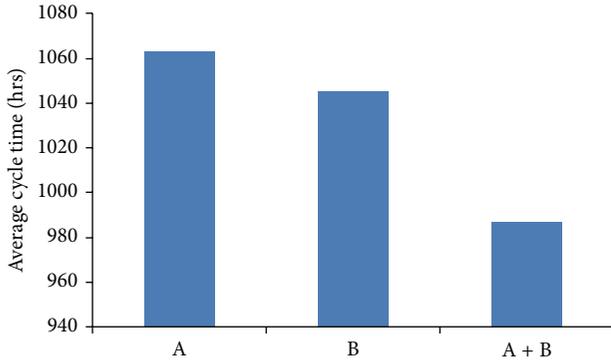


FIGURE 6: The effects of different treatments (A: the new way of slack diversification; B: forming the BPN ensemble).

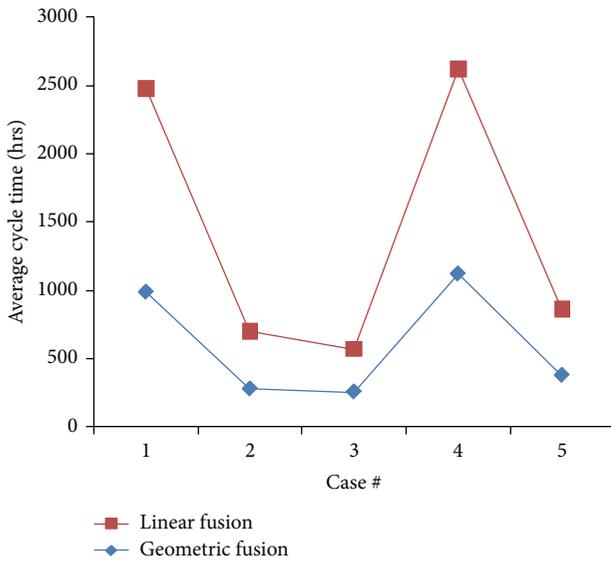


FIGURE 7: The performance of different fusion mechanisms.

The effectiveness of the proposed FCM-BPN-ensemble-2f-biFS was evaluated with some experiments. According to the experimental results, the following happened.

- (1) The proposed FCM-BPN-ensemble-2f-biFS outperformed the eleven existing methods/rules in reducing the average cycle time and cycle time standard deviation simultaneously.
- (2) To diversify the slacks of jobs, maximizing the geometric mean of the neighboring distances of slacks is a better choice than maximizing the standard deviation of slacks.
- (3) Improving the accuracy of the remaining cycle time estimation was shown to be a more effective way to the improvement of the scheduling performance, while slack diversification has its limits.

However, this study has its limits that can only be solved by applying it to an actual wafer fabrication factory. In addition, developing different versions of the proposed rule for

bottleneck and nonbottleneck machines can be considered in future studies. Further, to optimize the values of the parameters, other search algorithms that reach near-optimal solutions in less time are to be developed.

Abbreviations

j :	Job index; $j = 1 \sim n$
u :	Step number
CR_{ju} :	The critical ratio of job j at step u ; $j = 1 \sim n$
CT_j :	The cycle time of job j
CTE_j :	The estimated cycle time of job j
DD_j :	The due date of job j
R_j :	The release time of job j
$RCTE_{ju}$:	The estimated remaining cycle time of job j from step u
RPT_{ju} :	The remaining processing time of job j from step u
SCT_{ju} :	The step cycle time of job j until step u
SK_{ju} :	The slack of job j at step u
SKM_{ju} :	The slack of job j at step u in FSMCT and its variants
SKV_{ju} :	The slack of job j at step u in FSVCT and its variants
t :	The current time
TPT_j :	The total processing time of job j
λ :	Mean release rate
h_l :	The output from hidden-layer node l , $l = 1 \sim L$
w_{pl}^h :	The connection weight between input node p and hidden-layer node l , $p = 1 \sim P$; $l = 1 \sim L$
w_l^o :	The connection weight between hidden-layer node l and the output node
x_p :	Inputs to the BPN, $p = 1 \sim P$
θ_l^h :	The threshold on hidden-layer node l
θ^o :	The threshold on the output node.

Acknowledgment

This work was supported by the National Science Council of Taiwan.

References

- [1] C. N. Wang and C. H. Wang, "A simulated model for cycle time reduction by acquiring optimal lot size in semiconductor manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 34, no. 9-10, pp. 1008-1015, 2007.
- [2] T. Chen and Y. C. Lin, "A fuzzy-neural fluctuation smoothing rule for scheduling jobs with various priorities in a semiconductor manufacturing factory," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 17, no. 3, pp. 397-417, 2009.
- [3] T. Chen and Y. C. Wang, "A nonlinear scheduling rule incorporating fuzzy-neural remaining cycle time estimator for scheduling a semiconductor manufacturing factory—a simulation study," *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 110-121, 2009.

- [4] D. A. Koonce and S. C. Tsai, "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule," *Computers and Industrial Engineering*, vol. 38, no. 3, pp. 361–374, 2000.
- [5] B. W. Hsieh, C. H. Chen, and S. C. Chang, "Scheduling semiconductor wafer fabrication by using ordinal optimization-based simulation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 599–608, 2001.
- [6] H. J. Yoon and W. Shen, "A multiagent-based decision-making system for semiconductor wafer fabrication with hard temporal constraints," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 83–91, 2008.
- [7] Y. Harrath, B. Chebel-Morello, and N. Zerhouni, "A genetic algorithm and data mining based meta-heuristic for job shop scheduling problem," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 7, pp. 280–285, October 2002.
- [8] K. Sourirajan and R. Uzsoy, "Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication," *Journal of Scheduling*, vol. 10, no. 1, pp. 41–65, 2007.
- [9] A. K. Gupta and A. I. Sivakumar, "Job shop scheduling techniques in semiconductor manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 27, no. 11-12, pp. 1163–1169, 2006.
- [10] K. Altendorfer, B. Kabelka, and W. Stöcher, "A new dispatching rule for optimizing machine utilization at a semiconductor test field," in *Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC '07)*, pp. 188–193, Stresa, Italy, June 2007.
- [11] H. Zhang, Z. Jiang, and C. Guo, "Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology," *International Journal of Advanced Manufacturing Technology*, vol. 41, no. 1-2, pp. 110–121, 2009.
- [12] Z. Cao, Y. Peng, and Y. Wang, "A drum-buffer-rope based scheduling method for semiconductor manufacturing system," in *Proceedings of the 7th IEEE International Conference on Automation Science and Engineering (CASE '11)*, pp. 120–125, Trieste, Italy, August 2011.
- [13] Y. H. Lee, C. T. Chang, D. S. H. Wong, and S. S. Jang, "Petri-net based scheduling strategy for semiconductor manufacturing processes," *Chemical Engineering Research and Design*, vol. 89, no. 3, pp. 291–300, 2011.
- [14] I. L. Wang, Y. C. Wang, and C. W. Chen, "Scheduling unrelated parallel machines in semiconductor manufacturing by problem reduction and local search heuristics," *Flexible Services and Manufacturing Journal*, vol. 25, no. 3, pp. 343–366, 2013.
- [15] T. Chen, "Fuzzy-neural-network-based fluctuation smoothing rule for reducing the cycle times of jobs with various priorities in a wafer fabrication plant: a simulation study," *Proceedings of the Institution of Mechanical Engineers B*, vol. 223, no. 8, pp. 1033–1043, 2009.
- [16] H. Hu, L. Zhen, Z. Sun, and H. Zhang, "A multi-stage fluctuation smoothing method for multiple bottlenecks in wafer fabrication," *International Journal of Advanced Manufacturing Technology*, 2013.
- [17] C. C. Chern and Y. L. Liu, "Family-based scheduling rules of a sequence-dependent wafer fabrication system," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 1, pp. 15–24, 2003.
- [18] T. Chen, "A tailored non-linear fluctuation smoothing rule for semiconductor manufacturing factory scheduling," *Proceedings of the Institution of Mechanical Engineers I*, vol. 223, no. 2, pp. 149–160, 2009.
- [19] T. Chen, "Intelligent scheduling approaches for a wafer fabrication factory," *Journal of Intelligent Manufacturing*, vol. 23, no. 3, pp. 897–911, 2012.
- [20] R. M. Dabbas and J. W. Fowler, "A new scheduling approach using combined dispatching criteria in wafer fabs," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, pp. 501–510, 2003.
- [21] R. M. Dabbas, H. N. Chen, J. W. Fowler, and D. Shunk, "A combined dispatching criteria approach to scheduling semiconductor manufacturing systems," *Computers and Industrial Engineering*, vol. 39, no. 3-4, pp. 307–324, 2001.
- [22] T. C. Chiang, Y. S. Shen, and L. C. Fu, "A new paradigm for rule-based scheduling in the wafer probe centre," *International Journal of Production Research*, vol. 46, no. 15, pp. 4111–4133, 2008.
- [23] T. Chen and Y. C. Wang, "A bi-criteria nonlinear fluctuation smoothing rule incorporating the SOM-FBPN remaining cycle time estimator for scheduling a wafer fab—a simulation study," *International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5-8, pp. 709–721, 2010.
- [24] T. Chen, Y. C. Wang, and Y. C. Lin, "A bi-criteria four-factor fluctuation smoothing rule for scheduling jobs in a wafer fabrication factory," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 10, pp. 4289–4303, 2010.
- [25] Y. D. Kim, D. H. Lee, J. U. Kim, and H. K. Roh, "A simulation study on lot release control, mask scheduling, and batch scheduling in semiconductor wafer fabrication facilities," *Journal of Manufacturing Systems*, vol. 17, no. 2, pp. 107–117, 1998.
- [26] R. M. Dabbas, J. W. Fowler, D. A. Rollier, and D. McCarville, "Multiple response optimization using mixture-designed experiments and desirability functions in semiconductor scheduling," *International Journal of Production Research*, vol. 41, no. 5, pp. 939–961, 2003.
- [27] X. Li and O. Sigurdur, "Data mining for best practices in scheduling data," in *Proceedings of the IIE Annual Conference and Exhibition*, p. 853, May 2004.
- [28] T. Chen, "An optimized tailored nonlinear fluctuation smoothing rule for scheduling a semiconductor manufacturing factory," *Computers and Industrial Engineering*, vol. 58, no. 2, pp. 317–325, 2010.
- [29] T. Chen, "A nonlinear programming and artificial neural network approach for optimizing the performance of a job dispatching rule in a wafer fabrication factory," *Applied Computational Intelligence and Soft Computing*, vol. 2012, Article ID 471973, 9 pages, 2012.
- [30] T. Chen, "Incorporating fuzzy c-means and a back-propagation network ensemble to job completion time prediction in a semiconductor fabrication factory," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2153–2168, 2007.
- [31] S. Z. Zhao, Y. Cao, and Q. F. Tian, "Urban rail transit scheduling decision based on information fusion," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 41, supplement 1, pp. 85–88, 2011.
- [32] G. Lebre, G. Yao, M. Ait-Ahmed, and T. Tang, "A gain-scheduling and intelligence fusion method for fault-tolerant control," *IFAC Proceedings*, vol. 6, no. 1, pp. 1258–1263, 2006.
- [33] X. Li, H. Kang, and H. H. Chen, "Sensing workload scheduling in hierarchical sensor networks for data fusion applications," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '07)*, pp. 214–219, August 2007.

- [34] S. C. H. Lu, D. Ramaswamy, and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pp. 374–388, 1994.
- [35] S. D. Ashok and G. L. Samuel, "Least square curve fitting technique for processing time sampled high speed spindle data," *International Journal of Manufacturing Research*, vol. 6, no. 3, pp. 256–276, 2011.
- [36] T. Chen, "Dynamic fuzzy-neural fluctuation smoothing rule for jobs scheduling in a wafer fabrication factory," *Proceedings of the Institution of Mechanical Engineers I*, vol. 223, no. 8, pp. 1081–1094, 2009.
- [37] Y. C. Wang, T. Chen, and C. W. Lin, "A slack-diversifying nonlinear fluctuation smoothing rule for job dispatching in a wafer fabrication factory," *Robotics and Computer Integrated Manufacturing*, vol. 29, no. 3, pp. 41–47, 2013.
- [38] S. F. Liu and F. L. Chen, "A data clustering model for wafer yield loss in semiconductor manufacturing," *Journal of the Chinese Institute of Industrial Engineers*, vol. 21, no. 4, pp. 328–338, 2004.
- [39] C. H. Wang, "Recognition of semiconductor defect patterns using spatial filtering and spectral clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1914–1923, 2008.
- [40] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

